

# Detailed Syllabus of C Programming

Semester: I

**Course Title: C Programming**

**Course no: BIT102**

**Nature of course: Theory + Lab**

**Full Marks: 60+20+20**

**Pass Marks: 24+8+8**

**Credit hours: 3**

## Course Description:

This course familiarizes students with basic principles of programming. It introduces structured programming paradigm using a high level language called C. It covers the concept of problem solving techniques, program design, and basic elements of C along with the detailed concept of operators, statements, arrays, functions, pointers, structures and file handling.

## Course Objective:

The main objective of this course is to acquaint the students with good program design through structured programming paradigm for developing program for specific tasks using C Programming Language as well as to present the syntax and semantics of the C language.

## Course Contents:

Unit 1: Introduction (3 Hrs.)	Teaching Methodology	Teaching Hours
History and Advantages of C, Problems Analysis, Algorithm and Flowchart ( <i>introduction and characteristics of algorithm, introduction and symbols of flowchart start/stop, read/print, processing statement, condition check, direction of flow</i> ), Structure of a C Program ( <i>preprocessor directive, #include and #define directives, header files and library files</i> ); Writing, compiling, Debugging, Executing and Testing a C Program in windows and Linux/Unix like environment ( <i>compiler, integrated development environment, compiling and linking, compiler error, linker error, and run-time error</i> )	Lecture/Lab	3Hrs.
<b>Unit 2: Elements of C (3 Hrs.)</b>		
C Tokens ( <i>keywords, identifiers, operators, constants, and special symbols</i> ); C Character Sets ( <i>letters, digits, special characters and white spaces</i> ); Data types ( <i>Basic, Derived, and User Defined</i> ), Constants and Variables; Expression, Statements and Comments; Escape sequences and Delimiters	Lecture/ Lab	3 Hrs.
<b>Unit 3: Input/output Functions (2 Hrs.)</b>		
Conversion Specifiers; I/O functions; Formatted and Unformatted I/O ( <i>sacanf, printf, getchar, putchar, getch, getche, gets, puts, putch, getc, putc</i> )	Lecture/ Lab	2 Hrs.
<b>Unit 4: Operators and Expressions (4 Hrs.)</b>		
Arithmetic operators; Relational operators; Logical operators; Assignment operators; Type conversion in assignment; Increment and decrement operators;	Lecture/Lab	4 Hrs.

Ternary operator (conditional operator); Bitwise operator; Other operators (comma, sizeof); Expression evaluation (implicit and explicit type conversion; Operator precedence and associativity)		
<b>Unit 5: Control Structures (8 Hrs.)</b>		
Introduction control statements, Selection statements: if, if..else, if..else if ladder, nested if, switch case; break and continue, goto and labels	Lecture/ Lab	4 Hrs.
Iterative statement: For Loop, While Loop, Do while Loop, Nested Loop; The odd loop; Controlling the loop execution, Exit statement		4 Hrs.
<b>Unit 6: Arrays and Strings (5 Hrs.)</b>		
Introduction to Arrays; Initializing Arrays; The meaning of array indexing; One dimensional and Multidimensional Arrays (two dimensional only); String and Basic functions dedicated to string manipulation ( <i>reading and writing strings, null character, strcpy, strcmp, strcat, strlen,strupr, strlwr, strrev</i> )	Lecture/ Lab	5 Hrs.
<b>Unit 7: Functions (6 Hrs.)</b>		
Introduction and types of functions; Declaring, Defining and Calling functions; Arguments and Return Statement; Recursive functions; Function call by value and reference; Variables' scope, local variables and function parameters; Arrays as function parameter; Void as a parameter; Parameterizing the main function; External function and variables; Header files; Static variables; Register Variables	Lecture/ Lab	6 Hrs.
<b>Unit 8: The C Preprocessor (2 Hrs.)</b>		
Features of C Preprocessor; Macro Expansion; Macros with Arguments; Macros versus Functions; File Inclusion; Conditional Compilation; #if and #elif Directives; #undef Directive; #pragma Directive; The Build Process; Preprocessing; Compiling; Assembling; Linking; Loading	Lecture/ Lab	2 Hrs.
<b>Unit 9: Pointers ( 5 Hrs.)</b>		
Introduction of Pointers, declaration and initialization of pointer variables; An address, a reference, a dereference and the sizeof operator; Pointer to nothing (NULL); Pointer assignment; Pointer Arithmetic; Pointer as argument and Pointer as return values; Pointers vs. arrays; Dynamic memory allocation	Lecture/ Lab	5 Hrs.
<b>Unit 10: Structure and Unions (5 Hrs.)</b>		
Definition of Structure; Array of structures; Passing structure and array of structure to function; Pointers to structures and arrays of structures; Self-referential structures; Typedef; Table Lookup; Unions	Lecture/ Lab	5 Hrs.
<b>Unit 11: File Handling (2 Hrs.)</b>		
Files vs. streams; Header files needed for stream operations; Opening and closing a stream, open modes, errno variable; Reading and writing to/from a stream; Predefined streams: stdin, stdout and stderr; Stream manipulation: fgetc(), fputc(), fgets() and fputs() functions; Raw input/output: fread() and fwrite() functions; Random access to files	Lecture/ Lab	2 Hrs.

Text / Reference Books:

1. Let Us C, Yashavant P. Kanetkar
  2. Brian Kernighan and Dennis Ritchie, The C Programming Language
  3. Byron Gottfried, Programming with C, McGraw Hill Education
- Reference Books:

## **Laboratory work in C programming**

C Programming is the first programming course in BIT program. It builds the foundation on how to write a program using any high level language. This course is responsible for building the capability in students regarding development of good logic building and program which is essential throughout the BIT course and thereafter. Hence, Laboratory Works in this course emphasizes the verification of programming concepts learned in class. Each unit should include sufficient practical lab exercise. Further, 20% of the total marks is obtained from the practical. Students must spend 45 hours in lab activities with supervision of instructor.

Some important contents that should be included in lab exercises are as follows:

### **Unit 1,2,3,4:**

Using different data types available in C, perform arithmetic operations in C, perform formatted and unformatted input/output operations, perform character input/output operations.

Using relational operator, logical operator, assignment operator, ternary operator, and other operators. Evaluation of Expression to check operator precedence and associativity.

### **Unit 5:**

Create decision making programs using control statements like; if, if..else, if..else ladder, nested if, and switch cases.

Create programs using loops (for, while, do while, nested loops) and realize the differences between entry controlled and exit controlled loops.

### **Unit 6:**

Create, manipulate arrays and matrices (single and multi-dimensional), manipulate strings (character arrays) using various string handling functions.

### **Unit 7:**

Create user-defined functions with/without parameters or return type, create recursive functions, use function call by value and call by address, work with automatic, global and static variables.

### **Unit 8:**

Create programs to address macro, file inclusion, conditional compilation (#if, #ifdef, #endif, #ifndef, #else, #elif), building process

**Unit 9:**

Create programs that addresses pointer arithmetic, pointers and arrays, pointer and character strings, pointers and functions, pointer and structure, and dynamic memory allocation.

**Unit 10:**

Create and use simple structures, array of structures, nested structure. Passing structure and array of structure to function, concept of pointer to structure

**Unit 11:**

Create files that address random access and input/output operations in file, create files to keep records and manipulation of records etc.

Note: Students should be encouraged to create small project work integrating all of the above concepts.

## Model Questions

Level: BIT First Year/First Semester

Time: 3 Hrs

F.M: 60

Sub: Digital Logic (BIT103)

P.M: 24

---

Attempt any two questions. [2\*10=20]

1. Define array. Differentiate between 1D and 2D array. Write a program that stores 100 integers in an array and display them in ascending order. [1+3+6]
2. Define structure. Write syntax for defining and initializing structure? Write a program that stores details of N employees (E\_id, E\_Name, Salary), and display the details of those employees whose salary is less than 10000. [1+4+5]
3. Discuss the advantages of using functions in programming. Differentiate between recursive function and normal function. Write a program using call by reference to swap the values of two variables. [3+3+4]

Attempt any eight questions. [8\*5=40]

4. Define the basic structure of C program. Explain the Compilation and Execution of a C program? [2+3]
5. What are the uses of comments and escape sequences in a C program, explain with example? [5]
6. What are relational operators and arithmetic operators? Write a C code to check if a given number is odd or even. [2+3]
7. Define iterative statements. Write a program that using a loop to compute and prints the sum of squares of first 10 even natural numbers. [1+4]
8. How is string defined in C program? Write a program to check if a given string is a palindrome or not. [2+3]
9. Differentiate between switch case and else .. if ladder. Why is break statement used inside switch case? Explain briefly. [3+2]
10. Describe the concept of pointer and its arithmetic with suitable examples. [2+3]
11. Explain the syntax of fwrite() function. Write a program to create a file named "student.txt" and stores record of any five students given by user. [1+4]
12. Write short notes on: [2+3]
  - i) Formatted input/output
  - ii) C Preprocessors/directives